Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ **Hit List** ● **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Saturday, December 18, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L65	L64 and ("reverse star" or snowflake) near schema	38	L65
L64	L62 and (populat$ near datamart or populat$ and data near2 mart or data near warehouse or data with warehouse or "data warehouse")	528	L64
L63	L62 and (populat$ near datamart or populat$ and data near2 mart)	55	L63
L62	L61 and (metadata or meta with data)	4648	L62
L61	business and (data with base or database or data near2 base)	70991	L61
L60	717.clas.	7581	L60
L59	717/105	201	L59
L58	717/104	360	L58
L57	717/102	103	L57
L56	717/5	839	L56
L55	706/52	693	L55
L54	706.clas.	6303	L54
L53	705/44	946	L53
L52	705/39	1748	L52

L51	705/35	2246	L51
L50	705/30	1159	L50
L49	705/28	1772	L49
L48	705/26	5223	L48
L47	705/16	1112	L47
L46	705/14	3962	L46
L45	705/7	2219	L45
L44	705/5	888	L44
L43	705/1	5453	L43
L42	705.clas.	30759	L42
L41	707.clas.	24011	L41
L40	707/206	1043	L40
L39	707/201	2488	L39
L38	707/200	3668	L38
L37	707/104.1	4976	L37
L36	707/101	3681	L36
L35	707/100	5531	L35
L34	707/10	9618	L34
L33	707/1	7261	L33

DB=USPT; PLUR=YES; OP=OR

L32	5668987.pn.	1	L32
L31	5761657.pn.	1	L31
L30	5761657.pn.	1	L30
L29	5511186.pn.	1	L29
L28	5511186.pn.	1	L28
L27	5555409.pn.	1	L27
L26	5555409.pn.	1	L26
L25	5675785.pn.	1	L25
L24	5675785.pn.	1	L24
L23	6581062.pn.	1	L23
L22	5873096.pn.	1	L22
L21	5893075.pn.	1	L21
L20	5893075.pn.	1	L20
L19	6151601.pn.	1	L19
L18	6151601.pn.	1	L18
L17	6167405.pn.	1	L17
L16	5708828.pn.	1	L16
L15	5708828.pn.	1	L15
L14	5870746.pn.	1	L14
L13	5644740.pn.	1	L13
L12	5644740.pn.	1	L12

L11	5870746.pn.	1	L11
L10	5870746.pn.	1	L10
L9	5666528.pn.	1	L9
L8	5742738.pn.	1	L8
L7	5918232.pn.	1	L7
L6	5918232.pn.	1	L6
L5	6167405.pn.	1	L5
L4	6167405.pn.	1	L4
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L3	L2 and data near model	42	L3
L2	L1 and definit$	55	L2
L1	(reverse near star near schema or snowflake near schema)	61	L1

END OF SEARCH HISTORY

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

End of Result Set

□ Generate Collection Print

L4: Entry 1 of 1 File: USPT Dec 26, 2000

US-PAT-NO: 6167405
DOCUMENT-IDENTIFIER: US 6167405 A

TITLE: Method and apparatus for automatically populating a data warehouse system

DATE-ISSUED: December 26, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rosensteel, Jr.; Kenneth R.	Phoenix	AZ		
Guhr; Jerry T	Phoenix	AZ		
Picone; Joseph K.	Phoenix	AZ		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Bull HN Information Systems Inc.	Billerica	MA			02

APPL-NO: 09/ 067101 [PALM]
DATE FILED: April 27, 1998

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/102
US-CL-CURRENT: 707/102

FIELD-OF-SEARCH: 707/6, 707/101, 707/102, 395/785

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
□	5708828	January 1998	Coleman	395/785
□	5870746	February 1999	Knutson	707/101
□	5918232	June 1999	Pouschine et al.	707/103

OTHER PUBLICATIONS

"Data Warehousing An Introduction", by Grayce Booth, Groupe Bull Technical Update, Man/Jun. 1995, pp. 1-9, Copyright Jun. 1995.

"The Distributed Data Warehouse Solution", by Kirk Mosher and Ken Rosensteel, Groupe Bull Technical Update, May/Jun. 1995, pp. 11-18 Copyright Jun. 1995.
"Bull Warehouse Initiative", by Wayne W. Eckerson, Oct. 1996, Patricia Seybold Group, pp. 1-28, Copyright 1996.

ART-UNIT: 271

PRIMARY-EXAMINER: Amsbury; Wayne

ATTY-AGENT-FIRM: Driscoll; Faith F. Solakian; John S.

ABSTRACT:

A method and system for facilitating the creation of warehouse requests in a data warehouse system. During the design of the data warehouse tables, a repository tool is used for storing a number of new objects such as source and target databases, source and target tables and warehouse requests that are graphically defined and linked together by an administrator with the repository tool. The resulting visual design is so drawn so as to serve as input for each warehouse request to be generated. The administrator invokes a data replication component that operatively couples to the repository tool signaling that the warehouse request is to be implemented. The data replication component automatically creates the different subcomponents of the request by accessing various links stored by the repository tool and displays a visual representation of the subcomponents and their relationships to each other to the administrator. Thereafter, the replication component provides access to menu screens for enabling the administrator to visualize each of the subcomponents of the request and their properties for enabling modifications to be made to such subcomponents for completing configuration of all request subcomponents. Subsequently, the warehouse request can be scheduled to execute and populate the warehouse tables.

35 Claims, 13 Drawing figures

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

End of Result Set

Search Forms

Search Results

Help

User Searches

Preferences

Logout

□ Generate Collection Print

L1: Entry 61 of 61 File: USPT Dec 8, 1998

DOCUMENT-IDENTIFIER: US 5848408 A
TITLE: Method for executing star queries

Detailed Description Text (27):
STAR QUERY TRANSFORMATION WITH SNOWFLAKE SCHEMAS

Detailed Description Text (28):
A snowflake schema is a star schema in which the dimension tables themselves have dimension tables. For example, the store table 102 in FIG. 1 is a dimension table for the sales table 106. One of the columns of the store table 102 is "manager". The "manager" column of store table 102 may contain values from a primary key "manager" column of a "manager" table (not shown). The manager table could include additional information about each manager, such as the home address, social security number, and phone number of each manager.

Detailed Description Text (29):
In the present example, the store table 102 may be considered a first level dimension table, since it stores further information about a dimension of the fact table, while the manager table may be considered a second level dimension table because it stores further information about a dimension of a first level fact table. There may be any number of levels of dimension tables in a snowflake schema. Further, an N level dimension table may itself have any number of N+1 level dimension tables.

Detailed Description Text (30):
Star queries associated with snowflake schemas are similar to star queries for conventional star schemas except that star queries for snowflake schemas may include (1) constraints for columns of second or higher level dimension tables, and (2) join predicates that establish a correlation between a foreign key columns of lower level dimension tables and primary key columns of higher level dimension tables.

Detailed Description Text (32):
According to an embodiment of the invention, the star queries for snowflake schemas are transformed in the same manner as star queries for conventional star schemas in that subqueries are generated based on constraints specified for dimension tables. For example, the constraint "dim2.c7>100" would result in the generation of the subquery:

Detailed Description Text (34):
However, with snowflake schemas, a subquery generated based on a constraint specified for a higher level dimension table must include join predicates to connect the constrained dimension table back to a first level dimension table. In the exemplary query given above, the constraint "dim12.c4=20" on dim12 is a constraint on a second level dimension table. The join predicate that connects dim12 to a first level dimension table is "dim1.key12=dim12.key". Therefore, the subquery generated for the constraint "dim12.c4=20" must contain the join predicate "dim1.key12=dim12.key". Thus, the following subquery may be generated based on the

```
"dim12.c4=20" constraint:
```

Previous Doc Next Doc Go to Doc#